**3.1 Exploring Bias-Variance Trade-off Across Hypothesis Classes:**

The goal of the project is to examine dependence of bias-variance in different hypothesis classes, such as Decision Trees, K-Nearest Neighbour, Linear Regression, Polynomial Regression, and Kernel Methods on complexity of the model. The experiment involves generating synthetic datasets and manipulating the complexity by crafting multiple hypothesis classes, ranging from simpler to more complex models, encompassing H−n ⊆ H−n+1 ⊆ · · · ⊆ H0 ⊆ H1 ⊆ · · · Hn. Finally, the analysis the bias-variance trade-off curve across these models to derive conclusive insights.

A balance exists between a model's capacity to reduce bias and variance. Developing a deep understanding of these errors enables us to construct precise models and steer clear of the pitfalls of overfitting and underfitting.

While it's commonly thought to depend solely on model complexity, our study investigates whether other factors also play a role. We aim to understand the nuances of this trade-off, offering insights to enhance model selection and optimization.

**3.2 Methodology**

The methodology adopts an experimental approach to delve into the intricacies of the bias-variance trade-off within the context of,

(1) K-Nearest Neighbour

(2) Kernel Methods

(3) Linear Regression

(4) Polynomial Regression

(5) decision tree classifiers

It begins by framing the experiment around a central question: how does model complexity affect the balance between bias and variance in predictive performance? To address this question, a hypothesis class is defined, representing a continuum of model complexities, ranging from low to high.

Hypothesis class serves as the foundation for generating synthetic datasets that mimic the characteristics of real-world data while ensuring compatibility with various classifiers.

The experiment involves generating synthetic datasets and manipulating the complexity of hypothesis classes to observe how the bias and variance change. This can be tested by plotting bias-variance curves for each hypothesis class and analysing the results. We will utilize libraries such as NumPy or scikit-learn to generate synthetic datasets using functions from the hypothesis class.

**Dataset Synthesis:**

For KNN the Dataset is created by taking random numbers for feature X1, and X2 between a range for X and exponentiation of one variable and multiplication of the other added with some noise for Y. 1/k is chosen as a measure of complexity.

For SVM One dataset is created using 2 features X1, and X2 randomly between 0 and 1, and Y is defined using sum of X1+X2 >1 or not. The second dataset is moon shaped by making two interleaving half circles using Scikit learn library

For Linear regression the Dataset is created with 5 features for X using Scikit learn library for linear regression. Number of features used in model is chosen as measure of complexity.

For decision tree datasets involves careful consideration of key factors such as feature distributions, class separability. a synthetic regression dataset with specified characteristics, which can be used for further analysis.

For Polynomial regression the dataset is created by taking a evenly spaced numbers between a range for X and then taking a cubic polynomial dependant on X and adding noise to it for Y. Degree of polynomial is chosen as measure of complexity

Bias-variance decomposition is computed for each complexity level, facilitating a deep analysis of the bias-variance trade-off. By comparing how well each algorithm performs with different levels of complexity in the data, we can understand how they handle different situations. Graphs are plotted by libraries like Matplotlib or seaborn to plot bias-variance trade-off curves for various models

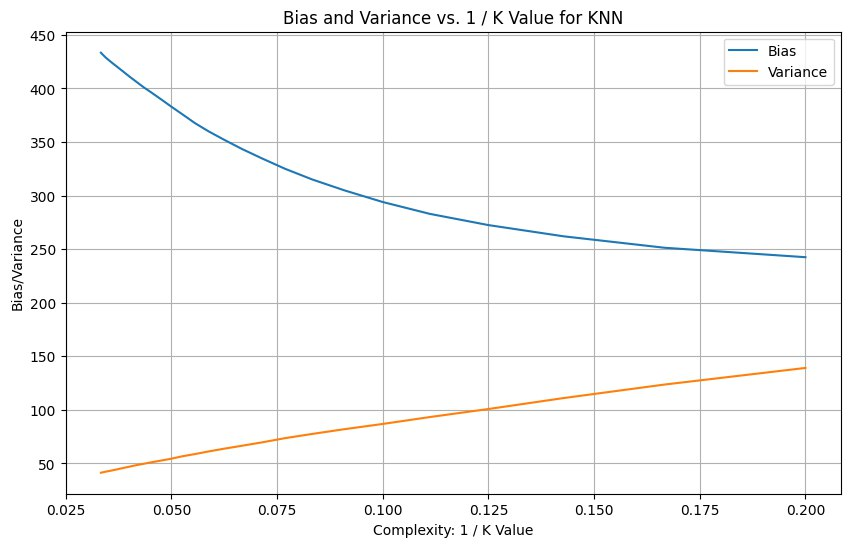
These visualizations provide an intuitive understanding of how each algorithm partitions the feature space and makes predictions, offering valuable insights into their strengths and weaknesses in handling different levels of complexity in the data.

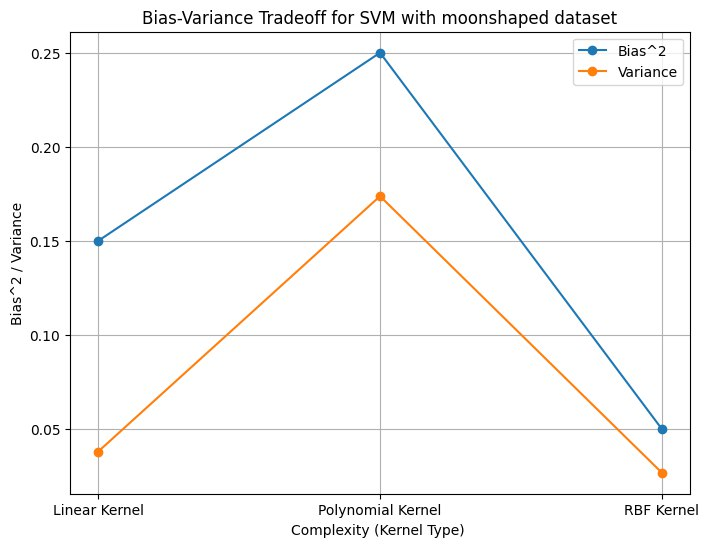
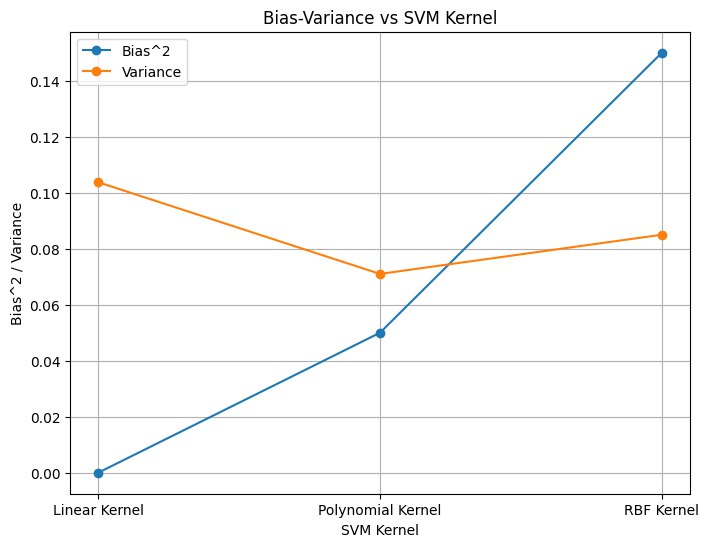
By integrating all these Python libraries into our project, we can efficiently generate synthetic datasets, manipulate model complexity, and visualize the bias-variance trade-off, facilitating a comprehensive analysis of the problem at hand.

Top of Form

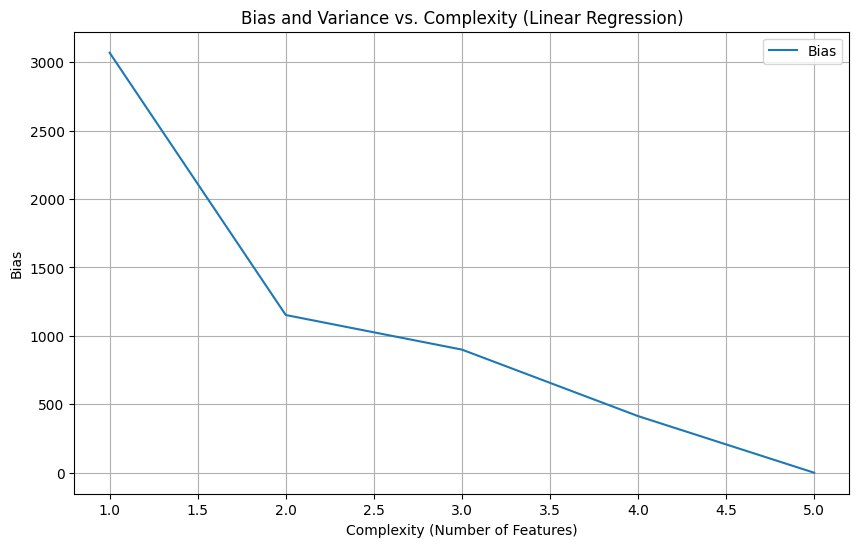
**3.3 Experimental Results and Validation**

**(1) K-Nearest Neighbour**

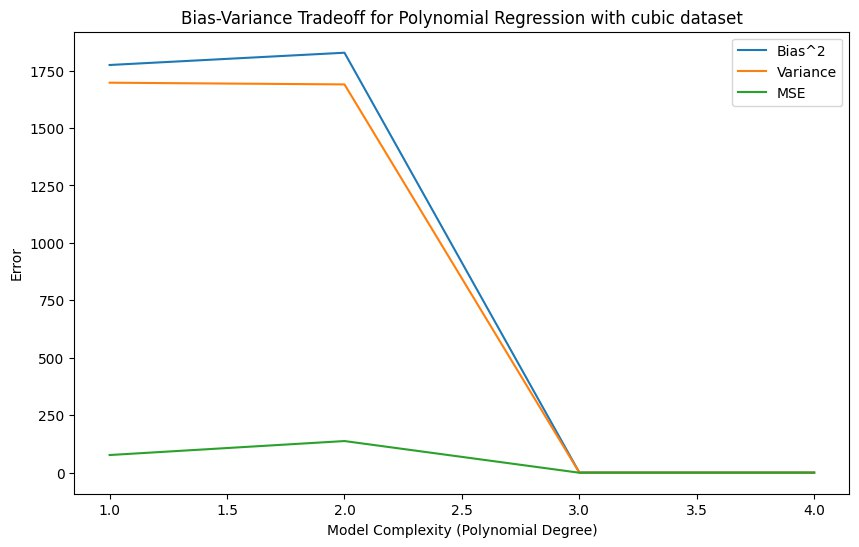
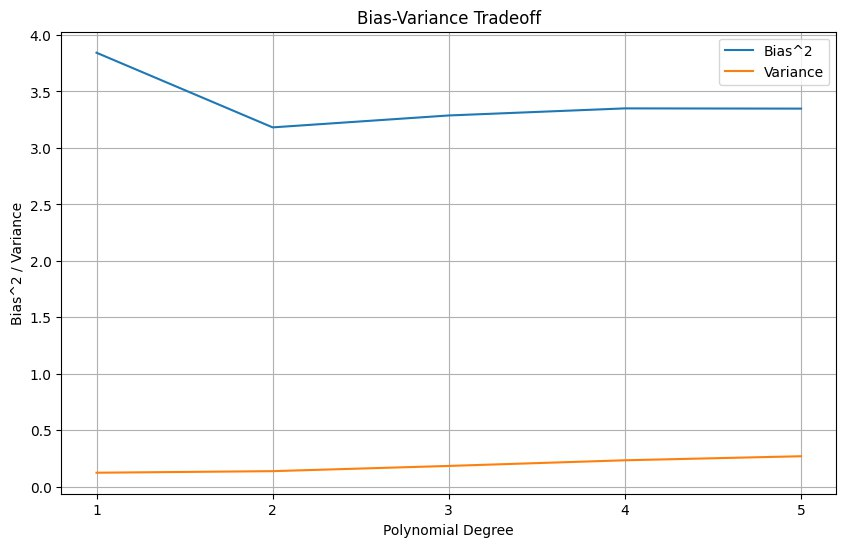


**(2) Kernel Methods,**

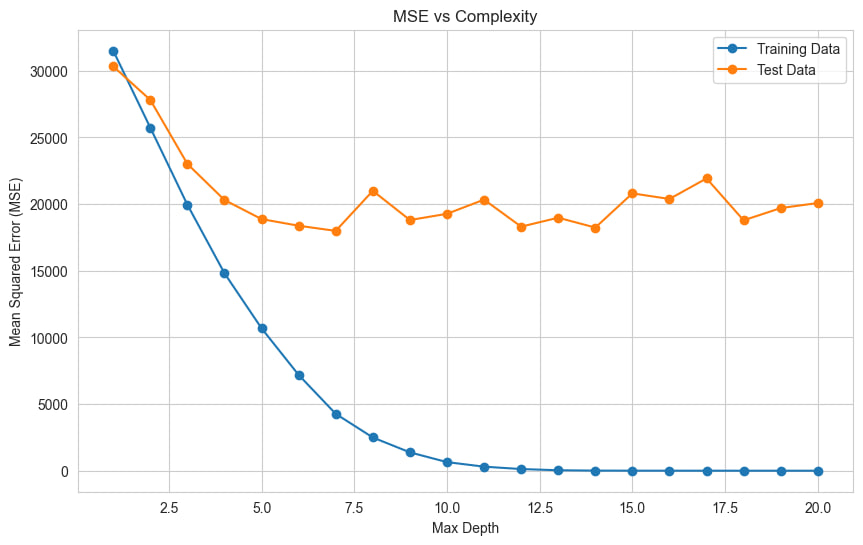
**(3) Linear Regression**



**(3) Polynomial Regression**



**(4) decision tree classifiers**

**3.3 Conclusion and Future Work**

In our study, we looked into how the complexity of a model affects its ability to balance between bias and variance in predictive modelling. The bias-variance trade-off does depend on the complexity of the model to some extent, but it's not the only factor. While model complexity plays a significant role in this trade-off, other factors such as the amount and quality of training data, the choice of features, and the regularization techniques used also influence the bias-variance trade-off. If the training data is noisy, biased, or incomplete, increasing model complexity may not improve performance.

Our study acknowledges that both the dataset and the hypothesis class can significantly impact error outcomes, emphasizing the need for careful consideration in model construction and analysis. Increasing complexity by adding more features can lead to the curse of dimensionality.

In the future, it's important to figure out which datasets and model types work best. By finding datasets where variance increases and bias decreases monotonically as the model gets more complex, we can learn what makes models perform best. This information can help us make better choices when picking models, improving how well they work in different situations.

**Name-**

1. **Shreyas Sinha (2022AAPS0799G)**
2. **Abhinav Pubey (2022AAPS1187G)**